Usage notes

126476_PixelRule.tiff ¬

I do not have a color NeXT computer. I do not have a color Macintosh. I think all my color conversions work, but am clearly unable to verify them *definitively*. If something is wrong, please send me electronic mail and explain exactly what the problem is.

This application tries to convert as much of the PICT data as it can. Nevertheless, the following aspects of PICT files are not fully converted (opcode numbers in square brackets, and in hexadecimal).

728019_paste.tiff ¬ 16 bit `direct' bitmaps *can not* be converted by Convert PICT at present. Some 32bit `direct' bitmaps may not convert as well. [9A, 9B]

235889_paste.tiff ¬ Some PicComments can be processed by Convert PICT now, but very few are fully implemented. [A0, A1].

787310_paste.tiff ¬ The inverting opcodes are not supported (invertRect, invertRRect, invertOval, invertArc, invertPolygon, and invertRegion). [33, 43, 53, 63, 73, 83]

994109_paste.tiff ¬ Text drawing modes are all ignored. Effectively, srcOr is used at all times. [05]

586847_paste.tiff ¬ Black and white bitmaps draw properly in srcOr or notSrcOr modes. If another mode is in use, or the bitmap is color, srcCopy is used. [90, 91, 98, 99]

585460_paste.tiff ¬ Pattern drawing supports these drawing modes (under a PS level 2 device): patCopy, patNotCopy, patOr, not PatOr, patXor, notPatXor (used for drawing lines and shapes). Foreground and background colors are used properly, I think. the remaining color modes default to a patCopy. [08]

315410_paste.tiff ¬ The hilite color opcodes and opColor are ignored. [1C, 1D, 1E, 1F]

610126_paste.tiff ¬ The headerop opcode is ignored. [0C00]

The following PICT instructions are not implemented because Apple's documentation is unclear and/or I have no working examples to dissect. In all cases, I'd love to get a working example that demonstrates their use!

572735_paste.tiff ¬ The opcode pnlochfrac is not supported, because I could not find documentation for the precise format of it's argument, and could not figure out what it does. [15]

285073_paste.tiff ¬ chextra is another opcode that I could not determine precisely what it does. [16]

896693_paste.tiff ¬ Similarly, the arguments to lineJustify seem pretty poorly defined, and I can't figure out what to do with them! [2D]

135121_paste.tiff ¬ Another case where I have no examples to help me figure out how to proceed is the fontName opcode [2C].

The following PICT details need special mention.

650386_paste.tiff ¬ On a PS Level 1 device, the Mac patterns are converted to shades of gray, rather than as a distinct pattern. These are rendered properly on a PS level 2 device.

904111_paste.tiff ¬ The color, non-8x8 patterns (Pattern type 1) may or may not be supported. I think the conversion code is in place, but I could not create or find a PICT file with one in it for me to test to make sure. If you find an example, I'd love to get a copy so I can make sure this works.

356401_paste.tiff ¬ The inner curves of round rectangle corners don't exactly match the curves that Apple uses (close enough for most needs, though)

113312_paste.tiff ¬ The use of the bounding rectangle in polygons and regions is not very well defined by Apple. In some unusual uses, even the Mac produces extremely unexpected results. I chose to interpret them as a clipping rectangle around the figure.

937241_paste.tiff ¬ There may be some boundary problems with the shapes (rect, oval, roundrect, arc, polygon, regions),

especially when the image is increased in size (e.g. when framed, arced objects boundaries are stroked, and are centered on the line, rather than within the line. So, they may appear to be off by half a unit in some circumstances)

909518_paste.tiff ¬ Bold text that is outlined or shadowed will end up with left borders that look odd.

711129_paste.tiff ¬ Shadowed characters that overlap their preceding neighbor will cast their shadow onto these chars, not behind.

956970_paste.tiff ¬ Opcodes not defined in Inside Mac V will be ignored.

66612_paste.tiff ¬ RGB color components are assumed to be 8 bits each, not 16 (which is right?)